









Matthew McRaven






Georgetown M.S. student
in Computer Science,
Research Assistant 

-  13888 SW 124th Ave,
Tigard OR 97223
-  mkm302@georgetown.edu
-  matthew.mcraven@gmail.com
-  (503) 964-2814
-  Matthew-McRaven
-  Matthew-McRaven










Languages

-  C/C++
-  QMake
-  LaTeX
-  Python
-  Bash
-  Java
-  x86 assembly basics

Frameworks & Libraries

-  Qt
-  POCO
-  Boost (namely Beast, ASIO)
-  Flatbufs & Protobufs
-  OpenCV

Tools

- Proficient in   
-  Git
-  Qt Creator
-  MS Visual Studio
-  Klee
-  MS Office Suite
-  Slack


Working Experience

- 2018 – Present **Research Programmer using Qt/C++** Collaboration with Pepperdine University
 - ★ Designed CISC processor (Pep/9) and a corresponding graphical processor simulator.
 - ★ Improved ISA simulation performance by 600× (written in C++).
 - Designed new processor architecture (and simulators at various abstraction levels) named Pep/10, featuring a linking macro-assembler.
 - Rewrote 4 applications to utilize common code base.
 - Created object code loader (in assembler) for Pep/10 operating system.
- 2019 – Present **Research Assistant using Python/Javascript** Georgetown University
 - ★ Helped design architecture for real-time network traffic capture and analysis using Python.
 - ★ Debugged issue in Python pipes causing lost data, prompting new traffic capture architecture.
 - Built visualization tool for network traffic flow using Python, Flask, and D3.js.
 - Automated deployment of Tor bridges using Docker.
- 2015 – 2019 **Software Developer using C++/Qt/Boost** SabrixTax LLC
 - ★ Prototyped multi-connection HTTP software router with dynamic routing rules for tax computations using C++/Boost.
 - ★ Automated cross-platform deployment of all SabrixTax applications.
 - Released multi-threaded HTTP client for replaying tax computations.
 - Compressed network traffic via Google's flatbuffers and protobufs.
 - Built performant C++ unit testing tool via Excel add-in (XLL).
 - Refactored existing Qt/C++ applications to use common code base.

Education

- 2019 – 2021 **M.S. in Computer Science** Georgetown University
(expected) Took (or currently taking) courses in Computer Architecture, Verification, Network Security. Designed chat client-server, website fingerprinting using Python. Current GPA 3.89 / 4.00
- 2015 – 2019 **B.S. in Computer Science/Mathematics** Pepperdine University
Graduated *Magna Cum Laude*. GPA 3.73 / 4.00
Took courses in Computer Systems, Organization, Operating Systems, Automata Theory.


Awards / Presentations

- 2019 – present **Computer Science Merit Scholarship** Georgetown University
Awarded to top 10% of incoming M.S. students.
- 2019 **Computer Science Student of the Year** Pepperdine University
Chosen as top C.S. student of the class of 2019.
- 2017 – 2019 **Northrop Grumman Scholarship** Pepperdine University
Awarded to top C.S. students at Pepperdine
- 2018 **Pep9Micro: Designing a Microcoded CPU**  SCCUR
Poster presentation on designing a new pedagogical processor.

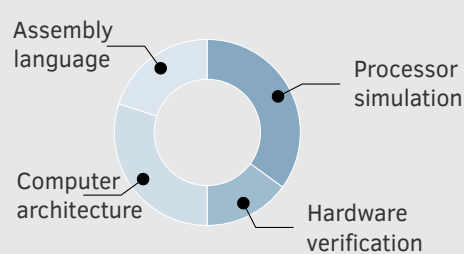
Extra-Curricular Activities

- 2015 – 2019 Child Program Volunteering Pacific Palisades Cavalry Church
- 2013 – 2014 Math Tutor Tigard High School
- 2012 – 2015 Martial Arts Instructor Family Martial Arts Academy

Matthew McRaven

Georgetown M.S. student
in Computer Science,
Research Assistant 

Research Interests






Developing Skills

-  Verilog
-  SoC Communication Architectures
-  Binary Code Analysis
-  PyTorch

Hobbies



Research / Projects

- 2019 – present **Pep/10**  Pepperdine University
Pep/10 is an in-development suite of tools to teach assembly language & computer organization concepts. It is a revision of Pep/9 featuring a macro-enabled assembly language.
- ★ Developed multi-pass assembler tool-chain to handle static linking and macro expansion.
 - Designed a recursive-descent parser for the new macro-enabled assembly language.
 - Created automated regression tests to evaluate the reliability of new assembler tool-chain.
 - Automated deployment/unit-testing via GitHub Actions.
- 2019 – present **Symbolic Verification of Pep/9**  Georgetown University
Pep9Milli is a hardware verification effort targeting the Pep/9 processor.
- ★ Verified that 75% of the Pep/9 CPU hardware/microcode adheres to its RTL specification.
 - Introduced a new hardware control language for Pep/9, *millicode*, that is easier to teach and debug than existing microcode.
 - Perform symbolic execution (using Klee) on millicode to verify processor correctness.
 - Paper publication describing verification effort is in progress.
- 2018 – present **Pep/9**  Pepperdine University
Pep/9 is a deployed tool suite to teach assembly language & computer organization concepts. Designed & built a hardware implementation of the Pep/9 processor.
- ★ Designed microcode/hardware implementation of the Pep/9 processor.
 - ★ Developed unified graphical microcode/ISA simulator, named Pep9Micro.
 - Implemented advanced assembly debugging features in ISA-level simulators, such as step into, over, & out of functions.
 - Created a terminal interface to the Pep/9 system, allowing for automated instructor grading of programming assignments.
 - Automated deployment on Windows, Mac OS, and Ubuntu.
 - Rewrote existing (Pep9 & Pep9CPU) applications to promote code reuse.